

整理番号

発送番号 009830

発送日 平成21年 1月27日 頁: 1/ 9

引用特許文獻

審判請求の番号	不服2006- 12852
(特許出願の番号)	(特願2002- 42589)
起案日	平成21年 1月26日
審判長 特許庁審判官	山崎 達也
請求人	インターナショナル・ビジネス・マシーンズ・コーポレーション 様
復代理人弁理士	正林 真之 様

引用文献5

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。 国内学会論文2000-00026-006

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

技術情報
TECHNICAL REPORT OF IEICE
CTR99-33, 1999-41 (1999-10)

OSI CCRを用いたUNIX用分散トランザクション処理方式

鈴木信雄 脇 英世

東京電機大学

東京都千代田区神田錦町2-2

あらまし 分散ネットワークにおける基本的な機能に分散トランザクション処理機能がある。現在、この機能を実現するシステムが多く開発され利用されているが、根本や用途等に統一がなく、アプリケーション開発者の負担となっている。一方、このような分散ネットワーク環境における通信方式の標準規格としてOSIがある。OSIは国際標準であることから概念や用語が統一され、将来の分散ネットワーク環境の共通の基盤を提供するものである。本稿では、汎用的な分散トランザクション処理機能をOSI標準のCCRを用いて実現するための方式について考察し、UNIX上で実現した結果について報告する。また、本実験においては、近年注目されているスレッドを用いて、分散トランザクション処理機能を提供するサーバプロセスを開発した。本稿では、その構造法についても報告する。

和文キーワード 分散トランザクション マルチスレッド環境

Distributed Transaction Processing Methodologies using OSI CCR for UNIX

Nobuo SUZUKI, Hideyo WAKI

Tokyo Denki University

Kanda-Nishiki-Cho 2-2, Chiyoda-ku, Tokyo 101, Japan

Abstract Distributed transaction processing is a basic facility for distributed networks. Although there are many systems which implement such facilities, those systems are not consistent. This situation causes application developers to be responsible for learning many systems. OSI (Open Systems Interconnection) can offer generalization of distributed transaction processing concepts and provides common basis. This paper proposes a method for distributed transaction processing using OSI CCR (Commitment, Concurrency and Recovery) in the UNIX environment. It also describes the construction of a multi-threaded server.

英文 key words distributed transaction multi-threaded server

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。 国内学会論文2000-00026-006

1. はじめに

現在、我々を取り巻くコンピュータ環境では、ほとんどのコンピュータはそれら単独で稼働しているのではなく、異なる環境を含む様々なコンピュータがLAN(Local Area Network)等のネットワークに接続されて利用されるのが一般である。そのため、最近では以前にも増してネットワーク機能の充実が重要となってきた。

このようなネットワーク環境の中で、分散配置された資源の更新などに有効な応用に分散トランザクション処理がある。分散トランザクション処理の概念は分散ネットワーク環境での基本的な機能であるにもかかわらず、現在、各社各様のシステムが開発されており、開発システムが異なる度によりアプリケーション開発者にとっては学習の負担が必要になってしまう。

一方、異機間の相互接続を目的としたOSI(開放型システム間相互接続)は、下位の通信規格の整備がほぼ完了し、現在は、RDA(リモートデータベース・アクセス)やOSI管理などの上位の応用層に関する標準化作業が活発に行われている。このOSIは、国際標準という性格から概念や用語が一般化され、将来の分散ネットワーク環境の共通の基盤を提供するものである。分散トランザクション処理の基本機能としてもCCR(コミットメント、同時性及び回復機能)が規定されている。そこで、筆者らは、現状のUNIX環境下における実用的な分散トランザクション処理機能を実現するための手段として、OSIS用層の機能単位であるCCRを用いることを提案し[6]、実験を試みた。本稿では、その実験手法について詳細に報告する。

また、マルチプロセッサの普及と高速化の要請から、UNIXに限らず多くのオペレーティングシステムでスレッドがサポートされている。今稿の実験では、マルチスレッド・サーバの実現についても検討及び実験を行っており[6]、この点についても報告する。

2. 分散トランザクション処理とOSI CCR

2.1. 分散トランザクション処理

従来のユーザが利用されるデータベース環

境を構築する上において必須となる技術にトランザクションがある。トランザクションは、ACIDと呼ばれる次のような4つの性質を持つ一連の操作を意味し、信頼性の高い資源更新処理などには重要な概念である。[3]

(1)原子性-atomicity: トランザクションとして定義された一連の操作は、完全に実行されるか、あるいは、全く実行されないかのいずれかである。

(2)一貫性-consistency: トランザクションの実行によってデータは矛盾のない状態から矛盾のない状態へ変化する。

(3)独立性-isolation: トランザクション途中の結果は他のトランザクションから参照することはできない。

(4)耐用性-durability: トランザクションが完了すると、その結果は永久的なもので、障害等によって失われることはない。

このようなトランザクションを資源がネットワーク上に分散配置された環境で実行するのが分散トランザクション処理である。

この分散トランザクション処理機能は、分散ネットワーク環境での基本的な機能であるにもかかわらず、各社各様のシステムにおいて概念や用語が異なっている。このような状況では開発システムが異なる度によりアプリケーション開発者にとっては学習の負担が生じてしまうという問題が出てくる。そこで、筆者らは、以下で述べるOSI CCRを用いることでUNIX上での実用的な分散トランザクション処理機能の実装手法について検討した。

2.2. OSI CCR

(1)CCRの概要

OSIは、コンピュータの異機間接続を目的として確立に保存しない通信規格を定義している。前述のような分散トランザクション処理は、分散ネットワーク上のアプリケーションにとって基本的な機能であるため、OSIにおいてもCCRやTP(トランザクション処理)として標準化されている。これら2つの標準のうち、TPは下位レベルでCCRを用い、複製ノードに対するトランザクション・サービスを提供する。これに対しCCRは、一対一のノード間での通信プロ

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。 国内学術論文2000-00026-006

トコルを規定しており、アプリケーションとして様々な形態のトランザクションを実装することが出来る。今回の検討では、(a)機能分散トランザクション処理機構の基本的なものに置かれているために柔軟性が高いこと、(b)システム全体のパフォーマンスに影響するプロトコル処理の複雑性が小さいことを考慮して、CCRを用いることとした。

また、このCCRは、国際標準であることから、異種環境拡張に限らず、今後の分散トランザクション処理のアプリケーション開発における共通の基盤となると考えられる。そのため、このCCRを利用することによって、分散トランザクション処理機構に関する概念や用語の統一を図り、汎用性の高いシステムを構築することが出来る。

(2)CCRの機能

CCRでは、アプリケーション間で行われるトランザクションを原子動作という名称で規定している。原子動作では、協調して動作するアプリケーションを(図1)のような原子動作木という木構造の関係として定義する。この木構造の上位と下位の間でCCRプロトコルを交換することによって原子動作を進めていく。また、CCRは、原子動作のコミットメント制御を行うために2相コミットメント・プロトコルを用いている。2相コミットメントでは、原子動作終了時に全ノードに処理が正常に終了したかどうかを尋ね、全てが肯定応答ならばコミットメント、一つでも否定応答があればロールバックの指示を出すことにより、全ノードの資源の一貫性を保証する。

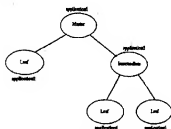


図1 CCRにおける原子動作木

また、原子動作に参加しているノードにおいては、障害が発生しても原子動作を維持することが必要である。そこで、CCRでは、原子動作データと呼ぶログ情報を保持することによって再処理を行うことが出来る。

図2CCRプロトコルのシーケンスを、表1にサービス定義を示す。



図2 CCRプロトコルのシーケンス

サービス名	機能	タイプ
C-BEGIN	原子動作開始	応答型
C-PREPARE	参加元からの問い合わせ	非応答型
C-READY	参加元からの応答	非応答型
C-COMMIT	参加元からの承諾	応答型
C-ROLLBACK	参加元からの拒否や指示	応答型
C-ABORT	原子動作の中断と指示	応答型

表1 CCRのサービス定義一覧

3. 実現手法

3.1.CCRの実装範囲

CCRでは、主に2相コミットメント機構に基づくコミットメント制御に関する機能を規定し

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものである。
取扱いにあたっては、著作権侵害とならないよう十分に注意してください。国内学会論文2000-00026-006

ており、同時性制御や互排制御の実現については、ほとんどが実装等に任されている。今回試作したシステムでは、CCRの中心的な機能であるコミットメント解除機能及び同時性制御機能を実現している。同時性制御機能としては、UNIXで提供されているロック機構を用いて資源(ファイル)全体をロックし、他のプロセスからのアクセスから保護することにより実現している。すなわち、今回の実装では、C-BEGIN、C-PREPARE、C-READY、C-COMMIT、C-ROLLBACKの各サービスを真似している。

3.2. 実現の基本方針

(1)分散トランザクション処理機能は独立したサーバとして実現する。

現在、多くのUNIXアプリケーションでは、クライアントサーバ方式のものが普及している。この方式を用いることにより、サーバの障害がコンピュータシステム全体の停止に至らないという高い耐障害性と、新しい機能の追加時にもサーバのみの変更だけで行えるという柔軟性を合わせ持つことができる。本システムにおいても分散トランザクション処理機能を実現するモジュールを独立したサーバプロセスとして生成し、クライアントからの要求により処理を行うこととする。

(2)サーバプロセス内はマルチスレッド・サーバとして実現する。

スレッドは、従来のプロセスに比べて起動と消滅に要する時間が短く、マルチプロセス環境での安全な並列化を実現できる機構である。一般に、トランザクション処理においては、処理時間の比較的に短い資源更新要求が短時間の内に多数発生するような状況が考えられる。このような状況に対してスレッドは特に有効であり、処理時間の短縮を図ることができる。そこで、本システムにおいては、分散トランザクション処理機能を実現するモジュールをマルチスレッド・サーバとして実現することにより、システム全体のパフォーマンスを向上させることとする。

(3)トランザクションサービス・インタフェースはCCRに倣った形式とする。

本実装では、分散トランザクション処理のためのノード間のプロトコルとしてCCRを用いて

いるが、アプリケーション開発者が実装に際してはクライアントとサーバ・モジュールにおけるトランザクションサービス・インタフェースについてもCCRに倣った形式とする。これにより、概念や用語だけでなく、API(アプリケーション・インタフェース)としての汎用性も実現することができる。故に各モジュールで使用するサービスの一覧を示す。

サービス名	説 明
TransOpen	トランザクション開始要求
TransSpecify	トランザクションの内部ノードへの参加要求
Begin	更新操作の開始要求
TransClose	トランザクション終了要求
Prepare	参加ノード間の問い合わせ要求
Commit	更新処理の完了
Cancel	コミットの取消
Rollback	コミットした取消
Rollback	ロールバックの要求通知

図2 トランザクション・サービス一覧

(4)下位の通信機構にはTCP/IPを用いる。

本来、CCR等のOSI応用層機能は7層から構成されるOSI参照モデルに倣ったネットワーク上に実現されるべきものである。しかし、今のOSIは普及の過渡期にあり、現実にはその仕様を満足する分散ネットワーク環境を実現することは困難である。そこで、現在、業界標準となっているTCP/IP (Transmission Control Protocol/Internet Protocol)を用いることで下位層の汎用性を図ることとする。すなわち、異なるノード間の通信にはコネクション型のInternetドメインでのソケットを使用し、自ノード内サーバとの通信にはUnixドメインのソケットを用いることとする。

(5)伝送されるプロトコル・データはASN.1形式に倣う。

OSI環境においては、伝送されるプロトコル・データはASN.1(Abstract Syntax Notation One)を用いて標準で定義されている。CCRも例外ではなく、各CCRプロトコル・データはASN.1によって定義されている。今回の実装で

本特許は、特許庁が著作権法第4条第2項第1号の規定により複製したものである。
取扱いにあたっては、著作権侵害とならないよう十分に注意すること。国内学術論文2000-00026-006

は下位層にTCP/IPを用いているが、将来のOSI環境への容易な移行を促すために、実際に伝送されるプロトコル・データはASN.1に従った定義をBER(Basic Encoding Rule)によって符号化したものを用いることとする。

3.3. システム構成

今回実現したシステムは、Sunワークステーション上のSunOS4.1を用いて構築されており、そのプロセス構成は図3に示すとおりである。また、以下に各プロセスにおける機能の概要を示す。

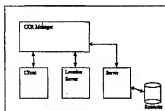


図3 プロセス構成

- (1)Client: 分散トランザクション処理機能を利用するユーザプロセスであり、CCR Managerプロセスに対してトランザクションの開始と終了、及び資源更新等の要求を行う。
- (2)CCR Manager: 分散トランザクション処理機能を提供するプロセスで、各ノード内に存在する。本プロセスは、トランザクションの遂行に必要な他ノードとのCCRプロトコル・データの交換や複数コネクションの維持機能等を有する。また、後述するようにマルチスレッド・サーバとして実現されており、パフォーマンスの向上を図っている。
- (3)Location Server: 他ノードとの通信時の位置追跡性を保証するためのプロセスで各ノード内に存在する。このプロセスを参照することにより、Clientプロセスからは、必要な資源を管理しているサーバのアドレスを知る必要がなく、サーバの名称のみを指定するだけで、該当するサーバと通信することができる。
- (4)Server: 実際の資源更新処理を行うプロセス。CCR Managerプロセスからのメッセージに

従ってコミットメント処理を遂行させると共に、UNIXの提供するロック機構により資源の同時性を保証する。

4. マルチスレッド環境の実現

スレッドは、同一プロセス内で複数の論理的な制御の流れを実行させることができる機構である。これは、複数クライアントに対するサービスを行うサーバなどに有効で、処理の高速化を図ることができる。今回の実現では、分散トランザクション処理機能を実現しているCCR Managerプロセスにおいてこの機構を利用して、ここで、マルチスレッド環境における通信アプリケーションを構築する上で前提となるのが、通信コネクション資源の管理である。通常、通信コネクションはプロセスが管理する資源であるためにスレッドが管理主体となるような機構が必要となる。

4.1. CCR Managerプロセス内スレッドの構成

SunOS4.1ではSunLWP(Lightweight process)ライブラリ[4]というユーザレベルのスレッド機構が提供されており、本システムではこれを使用している。図4はCCR Managerプロセス内でのスレッドの構成を示す。

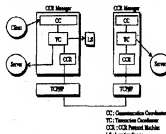


図4 CCR Manager内スレッドの構成

- (1)Communication Coordinator: Clientプロセスからの要求を受けてトランザクション毎にTransaction Coordinatorスレッドを生成する。また、複数ノードに渡るコネクション全体の管理を行

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものである。
取扱いにあたっては、著作権侵害とならないよう十分に注意してください。 国内学会論文2000-00026-006

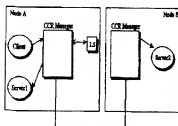


図6 システム構成例

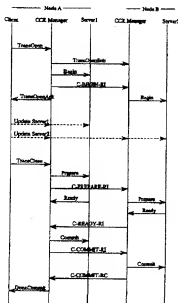


図7 シーケンス例

トランザクションの終了時には、Clientから
"TransClose"メッセージによってCCMに通知され
る。CCMは2個コミットメント・プロトコルに
対応するCCRプロトコルを各Serverの存在するノ

ード内のCMと連携することによってコミット
かロールバックかを決定する。コミットであれば
"DoneCommit"が、ロールバックであれば
"Rollback"がClientへ送渡される。

6. 考察

(1)分散ネットワーク環境における国際標準で
あるOSIの機能単位であるCCRを用いることに
よって、汎用的な分散トランザクション処理機
能を実現することができた。また、マルチス
レッド環境でのコネクション管理手法を工夫す
ることにより効率的な分散トランザクション処
理サーバを構築することができた。

(2)本システムでは、下位層としてTCP/IPを用
いているが、ネットワークで伝送されるCCRプ
ロトコル・データはASN.1の定義をBERによっ
て符号化されたオブジェクト列である。そこで、
下位層をOSI ACSE(Association Control Service
Element)以下のOSI環境に実行することにより、
比較的事業にOSI環境へ移行することが可
能である。

(3)本システムでは、CCRの持つ機能の内、コ
ミットメント耐障害機能と同時性耐障害機能を実
装した。しかし、回復耐障害機能もCCRの持つ重
要な機能の一つであり、分散トランザクション処
理においては必須の機能である。そこで、この
回復耐障害機能についても今後実現の検討を進め
ていく必要がある。実現に当たっては、コネク
ション資源と同じくログファイル等の資源もブ
ロセスが管理主体となるため、マルチスレッド
環境での管理手法について考慮しなければならない。

(4)OSI CCRは、OSIを用層の一機能単位であ
り、OSI参照モデルに基いたネットワーク環境
を前提として開発されたものである。しかし、
今後の実践により、既存のTCP/IPネットワ
ーク上でも実用的であることが示された。これは、
CCRのプロトコルが比較的小規模であること、
及び、トランスポート層に相当するTCP/IPの上
位に実装したことなどがその理由として挙げら
れる。

7. おわりに

本稿では、OSI CCRを用いたUNIX上での分

本複製物は、特許庁が著作権法第42条第2項第1号の規定により複製したものです。
取扱いにあたっては、著作権侵害とならないよう十分にご注意ください。 国内学会論文2000-00026-006

数トランザクション処理システムの実現法について述べた。本システムは、国際標準である OSI CCRを用いていることから、アプリケーション開発者に対する概念や用語等の統一を図ることができるという利点を持つ。また、トランザクション処理機能を提供するサーバプロセスをマルチスレッドサーバとし、この環境でのコネクション管理手法を工夫することによって効率的なサーバが実現できた。

参考文献

- [1]ISO/IEC: "IS 9804/9805 CCR", 1990
- [2]W.R.Stevens: "UNIX Network Programming", Prentice Hall, 1991
- [3]E.V.Krishnamurthy, V.K.Murthy: "Transaction Processing Systems", Prentice Hall, 1991
- [4]Jun Miyayama: "Programming Utilities and Libraries, Lightweight Processes", March 1990
- [5]Naer S. Barghouti, et. al.: "Concurrency Control in Advanced Database Applications", ACM Computing Surveys, Vol.23, No.3, Sep. 1991
- [6]鈴木嘉: "オープン環境を指向した分散トランザクション環境の構成法", 電子情報通信学会 1992年秋季大会, D-67, 1992.9
- [7]鈴木嘉: "OSI CCRを用いたUNIX用分散トランザクション処理環境の実現", 情報処理学会第45回全国大会, 4B-05, 1992.10